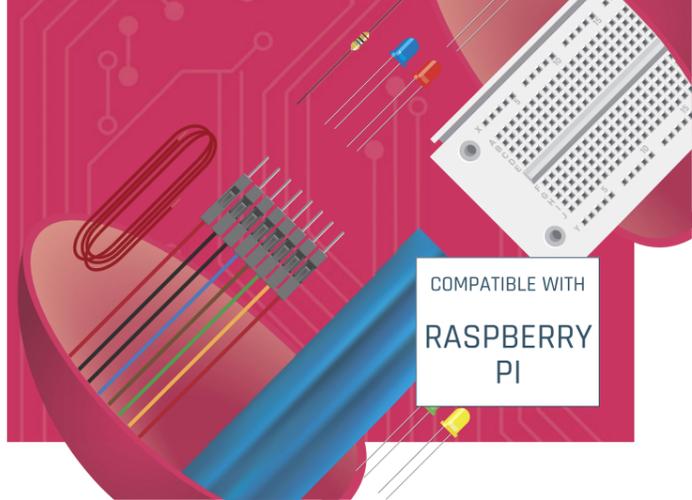




EI FÜR RASPBERRY PI

- Der ideale Einstieg in die Welt des Raspberry Pi
- Keine Vorkenntnisse nötig

Spannende Experimente mit dem Raspberry Pi



COMPATIBLE WITH
**RASPBERRY
PI**

WICHTIGER HINWEIS:

Die verwendeten Programme gibt es hier zum Download: conrad.de/technik-eier

Öffnen Sie die Webseite direkt mit dem vorinstallierten Browser auf dem Raspberry Pi und laden Sie die Zip-Datei in das Home-Verzeichnis /home/pi herunter. Starten Sie den Dateimanager auf dem Raspberry Pi. Dieser zeigt beim Start automatisch das Home-Verzeichnis an. Klicken Sie mit der rechten Maustaste auf die heruntergeladene Zip-Datei und wählen Sie im Kontextmenü **Hier entpacken**. Das Downloadarchiv enthält zudem die komplette Anleitung mit vielen weiteren Versuchen.

EI FÜR RASPBERRY PI

Mit einem normalen PC oder gar einem Notebook einfache Elektronik zu steuern, ist - auch wenn es nur ein paar LEDs sind - für Hobby-programmierer mit kaum vertretbarem Aufwand verbunden. Dem PC fehlen einfach die dafür notwendigen Schnittstellen. Außerdem ist das Windows-Betriebssystem denkbar ungeeignet dafür, mit Elektronik zu kommunizieren.

Der Raspberry Pi ist - obwohl es auf den ersten Blick gar nicht so aussieht - ein vollwertiger Computer. Vieles geht etwas langsamer, als man es von modernen PCs gewohnt ist, dafür ist der Raspberry Pi aber auch viel kleiner und vor allem billiger als ein PC.

Die Experimente in diesem Technik-Ei werden mit Scratch programmiert. Diese Programmiersprache ist auf dem Raspberry Pi vorinstalliert und überzeugt durch ihren geringen Einarbeitungsaufwand. Sie können ohne Vorkenntnisse sofort mit der Programmierung loslegen. Alle Experimente funktionieren mit dem Raspberry Pi 3 und dem Raspberry Pi 3 B+.

RASPBERRY PI VORBEREITEN

Um den Raspberry Pi in Betrieb zu nehmen, braucht man:

- USB-Tastatur und Maus
- HDMI-Kabel und Monitor
- Netzwerkkabel oder WLAN
- MicroSD-Karte mit Betriebssystem Raspbian (ab Version 2.8.2)
- Micro-USB-Handyladegerät als Netzteil (mindestens 2.000 mA)

Das Netzteil muss als Letztes angeschlossen werden, damit schaltet sich der Raspberry Pi automatisch ein. Es gibt keinen eigenen Ein-/Ausschalter.

Betriebssysteminstallation in kütze

Für alle, die ihren Raspberry Pi noch nicht mit der aktuellen Raspbian-Version betriebsbereit haben, hier die Systeminstallation in zehn Schritten:

1. NOOBS mindestens Version 2.8.2 von www.raspberrypi.org/downloads auf den PC herunterladen und Zip-Archiv auf die Festplatte entpacken.
2. Wurde die SD-Karte bereits benutzt, mit SD-Formatter im PC neu formatieren: www.sdcard.org/downloads/formatter_4. Dabei **Format Size Adjustment** einschalten (die SD-Karte muss mindestens 4 GB groß sein).
3. Alle Dateien und Unterverzeichnisse von NOOBS auf die SD-Karte kopieren.
4. SD-Karte aus dem PC nehmen, in den Raspberry Pi stecken und booten. Ganz unten **Deutsch** als Installationssprache wählen. Damit wird automatisch auch die deutsche Tastatur ausgewählt.
5. Das Häkchen beim vorausgewählten Raspbian-Betriebssystem setzen und oben links auf **Install** klicken. Nach Bestätigung einer Sicherheitsabfrage, dass die Speicherkarte überschrieben wird, startet die Installation, die einige Minuten dauert. Nach abgeschlossener Installation bootet der Raspberry Pi neu.
6. Seit der Version NOOBS 2.8.2 startet ein automatischer Konfigurationsassistent. Hier im ersten Dialogfeld auf **Next** klicken und dann Sprache und Zeitzone auswählen, falls sie nicht automatisch auf Deutsch gesetzt sind.



7. Im nächsten Schritt wird empfohlen, das Standardpasswort zu ändern, was für die Experimente in diesem Technik-Ei nicht unbedingt nötig ist.

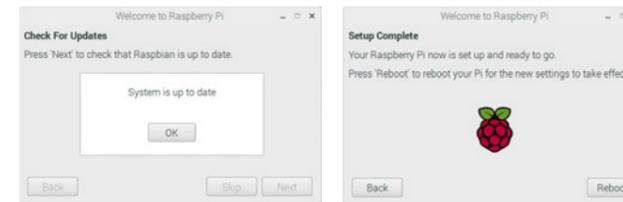


8. Bei **Select Wifi Network** das Netzwerk auswählen und das Passwort eingeben, bei Ethernetverbindung einfach auf **Skip** klicken.



9. Im nächsten Schritt wird gefragt, ob Updates heruntergeladen werden sollen. Je nach Anzahl und Größe der für das Betriebssystemimage verfügbaren Updates kann diese Installation bis zu einer Stunde dauern, kann aber auch mit **Skip** übersprungen werden.

10. Im letzten Schritt den Raspberry Pi neu booten.



NUM-LOCK-TASTE

Wie bei fast allen Linux-Systemen, ist auch beim Start von Raspbian der Ziffernblock standardmäßig ausgeschaltet. Drücken Sie die Num-Lock-Taste auf der Tastatur, um ihn einzuschalten.

DIE BAUTEILE

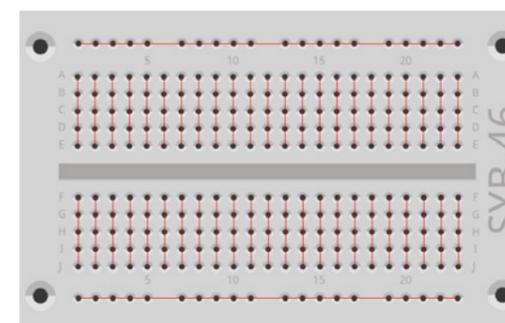
Diese Bauteile sind im Technik-Ei enthalten:

- 1 x Steckbrett
- 7 x LED mit Vorwiderstand
- 1 x Widerstand 20 MOhm
- 1 x Stück Knete
- 8 x GPIO-Verbindungskabel
- Schaltdraht

Steckbrett

Für den schnellen Aufbau elektronischer Schaltungen, ohne löten zu müssen, verwenden wir ein Steckbrett. Dort können elektronische Bauteile direkt in ein Lochraster gesteckt werden.

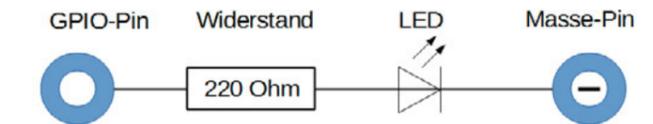
Bei diesem Steckbrett sind alle äußeren Längsreihen über Kontakte (X und Y) miteinander verbunden. Diese Kontaktreihen werden oft als Plus- und Minuspol zur Stromversorgung der Schaltungen genutzt. In den anderen Kontaktreihen sind jeweils fünf Kontakte (A bis E und F bis J) quer miteinander verbunden, wobei in der Mitte der Platine eine Lücke ist. Dort können größere Bauelemente eingesteckt und nach außen hin verdrahtet werden.



Die Verbindungen auf dem Steckbrett.

LEDs

LEDs (Deutsch: Leuchtdioden) leuchten, wenn Strom in Durchflussrichtung durch sie fließt. LEDs werden in Schaltungen mit einem pfeilförmigen Dreieckssymbol dargestellt, das die Flussrichtung vom Pluspol zum Minuspol oder zur Masseleitung angibt. Eine LED lässt in der Durchflussrichtung nahezu beliebig viel Strom durch, sie hat nur einen sehr geringen Widerstand. Um den Durchflussstrom zu begrenzen und damit ein Durchbrennen der LED zu verhindern, wird üblicherweise zwischen dem verwendeten GPIO-Pin und der Anode der LED oder zwischen der Kathode und dem Massepin ein 220-Ohm-Vorwiderstand eingebaut. Dieser Vorwiderstand schützt auch den GPIO-Ausgang des Raspberry Pi vor zu hohen Stromstärken. Die LEDs im Technik-Ei haben den Vorwiderstand bereits eingebaut und können daher direkt an die GPIO-Pins angeschlossen werden.



Schaltplan einer LED mit Vorwiderstand.

LED IN WELCHER RICHTUNG ANSCHLIESSEN?

Die beiden Anschlussdrähte einer LED sind unterschiedlich lang. Der längere ist der Pluspol, die Anode, der kürzere die Kathode. Einfach zu merken: Das Pluszeichen hat einen Strich mehr als das Minuszeichen und macht damit den Draht quasi etwas länger. Außerdem sind die meisten LEDs auf der Minusseite abgeflacht, vergleichbar mit einem Minuszeichen. Auch leicht zu merken: Kathode = kurz = Kante.

TIPP

Widerstände

Widerstände werden zur Strombegrenzung an empfindlichen elektronischen Bauteilen sowie als Vorwiderstände für LEDs verwendet. Die Maßeinheit für Widerstände ist Ohm. 1.000 Ohm entsprechen einem Kiloohm, abgekürzt kOhm. 1.000 kOhm entsprechen einem Megaohm, abgekürzt MOhm. Oft wird für die Einheit Ohm auch das Omega-Zeichen Ω verwendet.

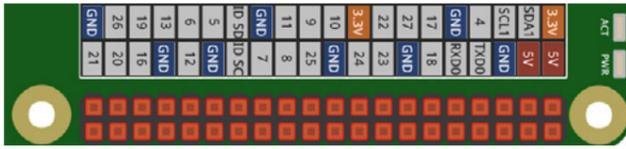
Die farbigen Ringe auf den Widerständen geben den Widerstandswert an. Mit etwas Übung sind sie deutlich leichter zu erkennen als winzig kleine Zahlen, die man nur noch auf ganz alten Widerständen findet. Die meisten Widerstände haben vier solcher Farbringe. Die ersten beiden Farbringe bezeichnen die Ziffern, der dritte einen Multiplikator und der vierte die Toleranz. Der Toleranzring ist meistens gold- oder silberfarben - Farben, die auf den ersten Ringen nicht vorkommen. Dadurch ist die Leserichtung immer eindeutig. Der Toleranzwert selbst spielt in der Digitalelektronik kaum eine Rolle. Die Tabelle zeigt die Bedeutung der farbigen Ringe auf Widerständen.

In diesem Technik-Ei ist ein 20-Mohm-Widerstand enthalten (rot-schwarz-blau). In welcher Richtung ein Widerstand eingebaut wird, ist egal. Bei LEDs dagegen spielt die Einbaurichtung eine wichtige Rolle.

Farbe	Widerstandswert in Ohm			
	1. Ring (Zehner)	2. Ring (Einer)	3. Ring (Multiplikator)	4. Ring (Toleranz)
Silber			$10^{-2} = 0,01$	$\pm 10 \%$
Gold			$10^{-1} = 0,1$	$\pm 5 \%$
Schwarz		0	$10^0 = 1$	
Braun	1	1	$10^1 = 10$	$\pm 1 \%$
Rot	2	2	$10^2 = 100$	$\pm 2 \%$
Orange	3	3	$10^3 = 1.000$	
Gelb	4	4	$10^4 = 10.000$	
Grün	5	5	$10^5 = 100.000$	$\pm 0,5 \%$
Blau	6	6	$10^6 = 1.000.000$	$\pm 0,25 \%$
Violett	7	7	$10^7 = 10.000.000$	$\pm 0,1 \%$
Grau	8	8	$10^8 = 100.000.000$	$\pm 0,05 \%$
Weiß	9	9	$10^9 = 1.000.000.000$	

GPIO-Verbindungskabel

Die farbigen Verbindungskabel haben alle auf einer Seite einen Stecker, auf der anderen Seite eine Steckbuchse, die auf einen GPIO-Pin des Raspberry Pi passt. Die Stecker werden in das Steckbrett gesteckt. Die programmierbaren GPIO-Pins auf dem Raspberry Pi haben Nummern, die Masse-Pins sind in der Abbildung mit GND gekennzeichnet.



Belegung der GPIO-Pins.

VORSICHTSMASSNAHMEN

Auf keinen Fall sollte man irgendwelche GPIO-Pins miteinander verbinden und abwarten ab, was passiert.

Nicht alle GPIO-Pins lassen sich frei programmieren. Ein paar sind für die Stromversorgung und andere Zwecke fest eingerichtet.

Einige GPIO-Pins sind direkt mit Anschlüssen des Prozessors verbunden, ein Kurzschluss kann den Raspberry Pi komplett zerstören. Verbindet man über eine LED zwei Pins miteinander, muss immer ein Schutzwiderstand dazwischengeschaltet werden. Eine Ausnahme bilden die LEDs mit eingebautem Vorwiderstand.

Für Logiksignale muss immer Pin 1 verwendet werden, der +3,3 V liefert und bis 50 mA belastet werden kann. Pin 6 ist die Masseleitung für Logiksignale.

Pin 2 und 4 liefern +5 V zur Stromversorgung externer Hardware. Hier kann so viel Strom entnommen werden, wie das USB-Netzteil des Raspberry Pi liefert. Diese Pins dürfen aber nicht mit einem GPIO-Eingang verbunden werden.

Schaltdraht

Mit dem Schaltdraht stellen Sie kurze Verbindungsbrücken her, mit denen Kontaktreihen auf der Steckplatine verbunden werden.

DIE PROGRAMMIERSPRACHE SCRATCH

Scratch ist auf dem Raspberry Pi im Menü unter 'Entwicklung' vorinstalliert und gilt als eine der am leichtesten zu erlernenden Programmiersprachen. Ein Programm wird aus puzzlesteinartigen Blöcken angeklickt, man braucht sich keine Befehlsnamen und Syntaxregeln zu merken.

Das neue Scratch 2

Seit der ersten Raspbian-Version war die Programmiersprache Scratch in der Version 1.x vorinstalliert. Für PCs gibt es seit einigen Jahren die neue Version Scratch 2 mit deutlich mehr Möglichkeiten. Damit lassen sich unter anderem eigene Funktionsblöcke erstellen.

Scratch 2 läuft auf dem PC online im Browser. Dafür ist allerdings mehr Rechenleistung erforderlich, als ein Raspberry Pi zurzeit bieten kann. Seit der Version NOOBS 2.4.0 ist im Raspbian-Betriebssystem eine Version von Scratch 2 vorinstalliert, die offline ohne Browser läuft und so mit der Leistung eines Raspberry Pi 3 problemlos auskommt. Mit Scratch 2 ist die Hardwaresteuerung über die GPIO-Schnittstelle wesentlich einfacher geworden. Allerdings werden einige wichtige Funktionen zur GPIO-Steuerung noch nicht unterstützt. Deshalb verwenden wir für alle Projekte in diesem Adventskalender weiterhin das bewährte Scratch 1.4.

Scratch vorbereiten

Klicken Sie oben links neben dem Scratch-Logo auf die Weltkugel und wählen Sie **Deutsch**. Die ausgewählte Sprache bleibt gespeichert, muss also nicht jedes Mal neu ausgewählt werden. Scratch 1.4 bietet eine Unterstützung für verschiedene



Scratch 1.4 auf Deutsch umschalten

Hardwarekomponenten am GPIO-Port, die bei jedem Programm über den Menüpunkt **Bearbeiten/Start GPIO server** einmal aktiviert werden muss. Dass die GPIO-Unterstützung aktiv ist, erkennen Sie daran, dass sich dieser Menüpunkt in **Stop GPIO server** ändert. Überprüfen Sie das bei jedem neuen Scratch-Programm.



Scratch 1.4 GPIO-Server starten

PROGRAMME ZUM DOWNLOAD

Die im Technik-Ei verwendeten Programme gibt es hier zum Download: conrad.de/technik-eier

Öffnen Sie die Webseite direkt mit dem vorinstallierten Browser auf dem Raspberry Pi und laden Sie die Zip-Datei in das Home-Verzeichnis `/home/pi` herunter.

Starten Sie den Dateimanager auf dem Raspberry Pi. Dieser zeigt beim Start automatisch das Home-Verzeichnis an. Klicken Sie mit der rechten Maustaste auf die heruntergeladene Zip-Datei und wählen Sie im Kontextmenü **Hier entpacken**.

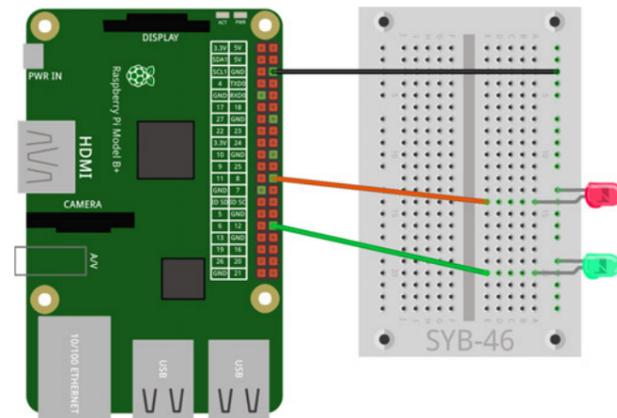
Das Downloadarchiv enthält diese Anleitung als PDF in Farbe, damit Sie auf den Schaltplänen die einzelnen Leitungen sowie die Abbildungen der Scratch-Programme besser erkennen können.

ZWEI LEDS BLINKEN ABWECHSELND

Das erste Experiment lässt zwei LEDs abwechselnd rot und grün leuchten. Gesteuert wird das Ganze über eine Endlosschleife in Scratch.

Bauteile

- 1 Steckbrett
- 1 LED rot mit Vorwiderstand
- 1 LED grün mit Vorwiderstand
- 3 GPIO-Verbindungskabel

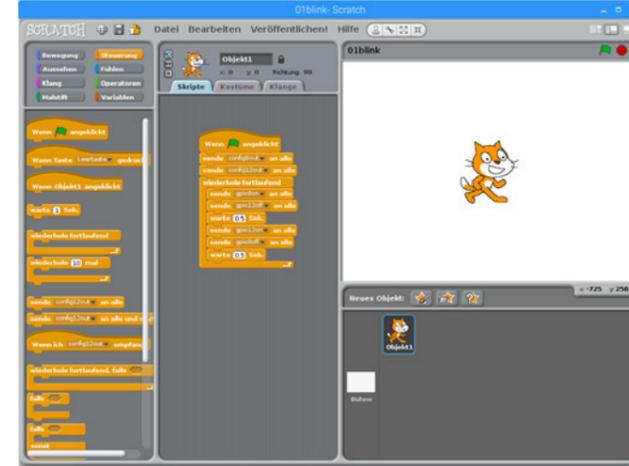


Zwei LEDs blinken am Raspberry Pi.

Achten Sie darauf, dass die LEDs richtig herum eingebaut sind. Die flache Seite ist in der Abbildung oben. Die Schaltungsaufbauten verwenden die Kontaktleiste an der einen Längsseite des Steckbretts als Massekontakt. Hier werden die Kathoden aller LEDs eingesteckt und über ein Kabel mit einem GND-Pin auf dem Raspberry Pi verbunden.

Das Programm

In Scratch braucht man beim Programmieren keinen Programmcode zu tippen. Die Blöcke werden einfach per Drag and drop aneinander gehängt. Die Blockpalette im linken Teil des Scratch-Fensters enthält, nach Themen geordnet, die verfügbaren Blöcke.



Dieses Scratch-Programm **01blink** steuert die beiden LEDs.

Sie können das Programm auf dem Bildschirm selbst zusammenbauen oder das Programm **01blink** aus dem Download verwenden. Wählen Sie dazu im Menü **Datei/Öffnen** und klicken Sie im nächsten Dialogfeld auf die Schaltfläche **pi**, um das persönliche Home-Verzeichnis auszuwählen, in dem die heruntergeladenen Programme liegen.

Klicken Sie in Scratch oben links auf das gelbe Symbol **Steuerung**. Dann werden in der Blockpalette links die Blöcke zur Steuerung angezeigt. Für dieses erste Programm brauchen wir nur diese gelben Blöcke.

Ziehen Sie die Blöcke, die Sie brauchen, einfach aus der Blockpalette in das Skriptfenster in der Mitte von Scratch.



Der Block **Wenn (Grüne Fahne) angeklickt** dient dazu, ein Programm zu starten. Die folgenden Skriptelemente werden ausgeführt, wenn man auf das grüne Fähnchen rechts oben in Scratch klickt. Der Block ist oben rund, passt also unter keinen anderen Block. Er muss immer als Erstes gesetzt werden.



Die GPIO-Befehle werden über den Scratch-Block **sende an alle** ausgegeben. Im Textfeld werden die jeweilige Pinbezeichnung und entsprechende Schlüsselwörter eingetragen. Klicken Sie dazu in das Textfeld im Block, wählen Sie **Neu/edit...** und geben Sie den Text ein. Am Anfang werden die GPIO-Pins 8 und 12 mit **config8out** und **config12out** als Ausgänge definiert. Jeder GPIO-Pin kann entweder Ausgang oder Eingang sein.



Eine **wiederhole fortlaufend**-Schleife sorgt dafür, dass die beiden LEDs endlos blinken, und zwar so lange, bis der Benutzer auf das rote Stopp-symbol oben rechts in Scratch klickt. Alle Blöcke innerhalb der Schleife werden wiederholt.



Nachdem die rote LED an Pin 8 eingeschaltet und die grüne an Pin 12 ausgeschaltet wurden, wartet das Programm eine halbe Sekunde. Dazu bietet Scratch einen eigenen Block **warte ... Sek.** an.



Danach werden auf die gleiche Weise die grüne LED an Pin 12 eingeschaltet und die rote an Pin 8 ausgeschaltet. Nach einer weiteren halben Sekunde wiederholt sich der Zyklus von vorne.

Programm speichern

Vergessen Sie nicht, das fertige Programm mit dem Diskettensymbol oben links zu speichern, um es später wieder verwenden zu können.

Sollen die LEDs schneller blinken, verkürzen Sie die Zeiten in den beiden **warte...Sek.**-Blöcken innerhalb der Schleife. Sollen sie langsamer blinken, verlängern Sie die Wartezeiten. Scratch verwendet wie viele amerikanische Programme den Punkt statt wie im Deutschen üblich das Komma. Eine halbe Sekunde Wartezeit schreibt man also 0.5 und nicht 0,5.

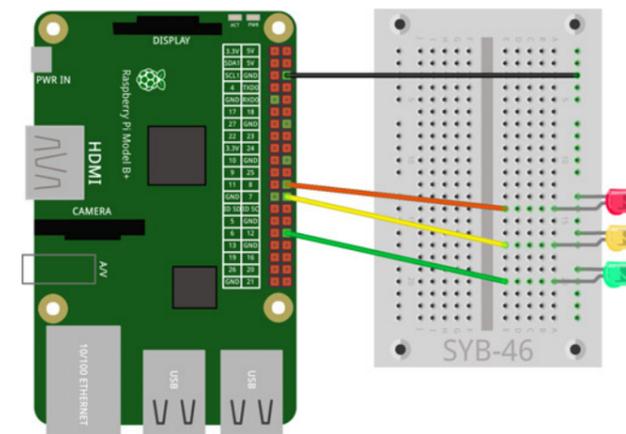
TIPP

ZWEI LEDS BLINKEN ABWECHSELND

Zwei LEDs ein- und wieder auszuschalten, mag im ersten Moment ganz spannend sein, aber dafür braucht man eigentlich keinen Computer. Eine Verkehrsampel mit ihrem typischen Leuchtzyklus von Grün über Gelb nach Rot und dann über eine Lichtkombination Rot-Gelb wieder zu Grün ist schon etwas spannender.

Bauteile

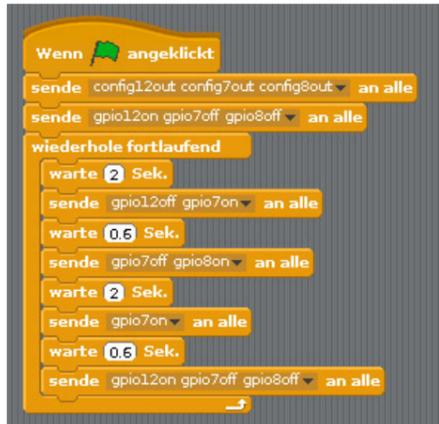
- 1 Steckbrett
- 1 LED rot mit Vorwiderstand
- 1 LED gelb mit Vorwiderstand
- 1 LED grün mit Vorwiderstand
- 4 GPIO-Verbindungskabel



Ampel aus drei LEDs.

Das Programm

Das Programm **02ampel01** enthält gegenüber dem vorherigen Programm nicht viel Neues.



Das Scratch-Programm **02ampel01** steuert die Ampel

Der erste **sende ... an alle** Block richtet die Pins 12, 7 und 8 als Ausgänge ein. Mehrere GPIO-Befehle können in einem Block gesendet werden.

Der zweite **sende ... an alle** Block schafft eine definierte Ausgangsstellung, falls die LEDs von einem früheren Programm noch leuchten. Die grüne LED am Pin 12 wird eingeschaltet, die gelbe LED an Pin 7 und die rote an Pin 8 werden ausgeschaltet.

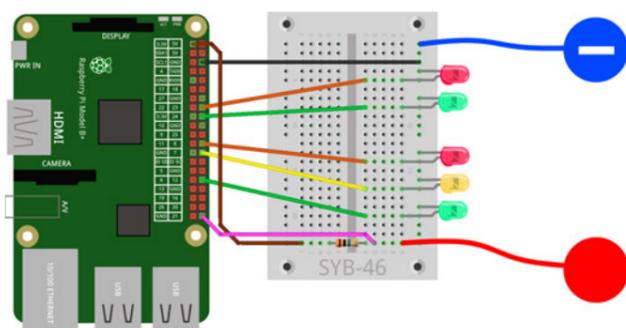
Die **wiederhole fortlaufend** Schleife beginnt mit der 2 Sekunden langen Grünphase. Danach erfolgt gelb und nach 0,6 Sekunden rot. Die Rotphase dauert wieder 2 Sekunden, anschließend schaltet die Ampel über rot/gelb auf grün. Danach startet die Schleife erneut. Dieser Zyklus wird wiederholt, bis der Benutzer auf das rote Stopp-Symbol klickt.

FUSSGÄNGERAMPEL MIT SENSORKONTAKT

Das nächste Experiment erweitert die einfache Ampelschaltung um eine Fußgängerampel, die während der Rotphase der Verkehrsampel eine Grünphase für Fußgänger anzeigt. Der Ampelzyklus läuft jetzt nicht mehr automatisch, sondern wird wie bei einer typischen Fußgängerampel durch Berühren einer Sensortaste gestartet.

Bauteile

- 1 Steckbrett
- 2 LED rot mit Vorwiderstand
- 1 LED gelb mit Vorwiderstand
- 2 LED grün mit Vorwiderstand
- 1 20-MOhm-Widerstand (Rot-Schwarz-Blau)
- 8 GPIO-Verbindungskabel
- 2 Knetkontakte

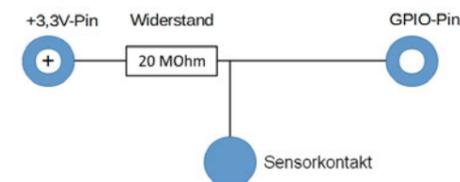


Fußgängerampel mit Sensorkontakt

Ampeln, Türöffner, Lichtschalter und Automaten werden heute oft mit Sensorkontakten gesteuert, die man nur zu berühren braucht. Taster, die wirklich gedrückt werden müssen, werden immer seltener.

Sensorkontakt aus Knete

Der als Eingang geschaltete GPIO-Pin ist über einen extrem hochohmigen Widerstand (20 MOhm) mit +3,3 V verbunden, sodass ein schwaches, aber eindeutig als High definiertes Signal anliegt. Ein Mensch, der nicht gerade frei in der Luft schwebt, ist immer geerdet und liefert über die elektrisch leitfähige Haut einen Low-Pegel. Berührt dieser Mensch jetzt einen Sensorkontakt, wird das schwache High-Signal von dem deutlich stärkeren Low-Pegel der Hand überlagert und zieht den GPIO-Pin auf Low-Pegel.



Wie hoch allerdings der Widerstand zwischen Hand und Masse wirklich ist, hängt von vielen Dingen ab, unter anderem von Schuhen und Fußböden. Barfuß im nassen Gras ist die Verbindung zur Masse der Erde am besten, aber auch auf Steinfußboden funktioniert es meistens gut. Holzfußböden isolieren stärker, Kunststoffbodenbeläge sind oft sogar positiv aufgeladen.

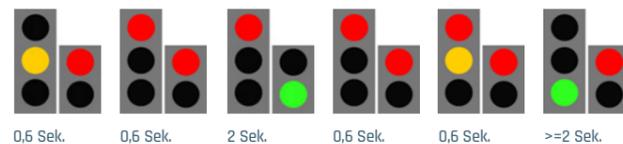
Knete leitet den Strom etwa so gut wie menschliche Haut. Sie lässt sich leicht in jede beliebige Form bringen, und ein Knetekontakt fasst sich viel besser an als ein einfaches Stück Draht. Die Fläche, mit der die Hand den Kontakt berührt, ist deutlich größer. So kommt es nicht so leicht zu einem „Wackelkontakt“. Stecken Sie ein Stück abisolierten Scheldraht in ein Stück Knete. Das andere Drahtende stecken Sie in das Steckbrett.

Sollte der Sensorkontakt nicht funktionieren, enthält die Schaltung einen zweiten Knetekontakt, der mit der Masseschiene des Steckbretts verbunden ist. Dieser ist in der Zeichnung mit einem Minuszeichen gekennzeichnet. Berühren Sie diesen und den eigentlichen Sensor gleichzeitig. Dann ist die Masseverbindung auf jeden Fall hergestellt.

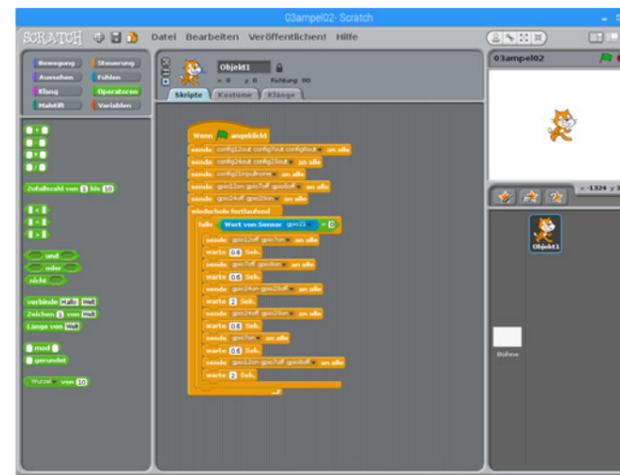
Das Programm

Das Programm **03ampel02** schaltet die Fußgängerampel. Klicken Sie auf das grüne Fähnchen, um das Programm zu starten. Die Verkehrsampel leuchtet grün, die Fußgängerampel rot, genau so, wie es echte Ampeln stundenlang tun würde, wenn kein Fußgänger kommt und auf den Knopf drückt.

Berühren Sie den roten Knetekontakt. Jetzt startet der Ampelzyklus, der in unserem Programm, wie auch bei einer echten Ampel, aus 6 unterschiedlichen Lichtmustern besteht, die unterschiedlich lange leuchten.



Mit dem letzten Lichtmuster - Fußgänger rot, Verkehrsampel grün - erreicht die Ampel wieder den Standardzustand. Das Programm muss allerdings dafür sorgen, dass auch dieser immer eine Mindestzeit eingehalten wird. Selbst wenn ständig Fußgänger auf den Knopf drücken, müssen die Autos auch mal fahren dürfen. In unserer Modellampel sind das 2 Sekunden, bei einer wirklichen Ampel natürlich deutlich länger.



Das Scratch-Programm **03ampel02** steuert die Fußgängerampel

Am Anfang werden die drei Pins 12, 7 und 8 für die Verkehrsampel sowie 24 und 23 für die Fußgängerampel als Ausgänge definiert. Der Pin 21 wird als Eingang für den Sensorkontakt eingerichtet. **...inpullnone** schaltet den am GPIO-Pin eingebauten Pull-down-Widerstand aus, da dieser die Funktion des Sensorkontakts beeinträchtigen würde.

Danach wird die Grundstellung der Ampel eingeschaltet, grün für die Autos, rot für die Fußgänger. Die anderen drei LEDs werden ausgeschaltet. Das Ausschalten ist hier am Anfang eigentlich nicht nötig. Es dient nur dazu, das Programm auf jeden Fall mit einem klar definierten Zustand zu starten.

Jetzt beginnt wie im letzten Programm eine **wiederhole fortlaufend** Schleife, die den Knetekontakt am GPIO-Pin 21 abfragt. Wenn dieser den Wert **0** liefert, wird er berührt und damit auf Low-Pegel gezogen.



Dazu verwenden wir den Scratch-Block **falls**, der ähnlich wie eine **if**-Abfrage in anderen Programmiersprachen funktioniert. Für die Abfrage selbst ist im **falls**-Block ein längliches Feld mit spitzen Enden vorgesehen. Hier muss ein Block aus der grünen Blockpalette **Operatoren** eingefügt werden. Wählen Sie den Block mit dem Gleichheitszeichen und ziehen ihn auf das Platzhalterfeld im Block **falls**.



Dieser Operator ist immer dann wahr, wenn die beiden Werte links und rechts des Gleichheitszeichens gleich sind. In unserem Fall soll der Wert des GPIO-Pins 21 dem Wert **0** entsprechen. Zur Abfrage von GPIO-Eingängen wird der Block **Wert von Sensor** aus der blauen Blockpalette **Fühlen** verwendet.



Ziehen Sie diesen Block auf das Platzhalterfeld links im grünen Gleichheitsoperator. Wählen Sie im Listenfeld des blauen Blockes den Sensor **gpio21** aus. Neben einigen vordefinierten Sensoren werden alle GPIO-Pins zur Auswahl angeboten, die als Eingang definiert sind. Klicken Sie jetzt einmal auf das grüne Fähnchen oben rechts, um das noch unfertige Programm zu starten. Damit werden die GPIO-Pins definiert und **gpio21** erscheint in der Auswahlliste. Halten Sie das Programm mit dem roten Stopp-Schild wieder an. Tragen Sie dann noch rechts im grünen Gleichheitsoperator den Wert 0 ein.

Jetzt werden nacheinander in Abständen von 0,6 Sekunden die unterschiedlichen Lichtmuster der Ampel geschaltet. Während die Fußgängerampel grün leuchtet, wird 2 Sekunden gewartet. Auch die Minstdauer der Grünphase der Verkehrsampel beträgt 2 Sekunden.

LEDS BLINKEN ZUFÄLLIG

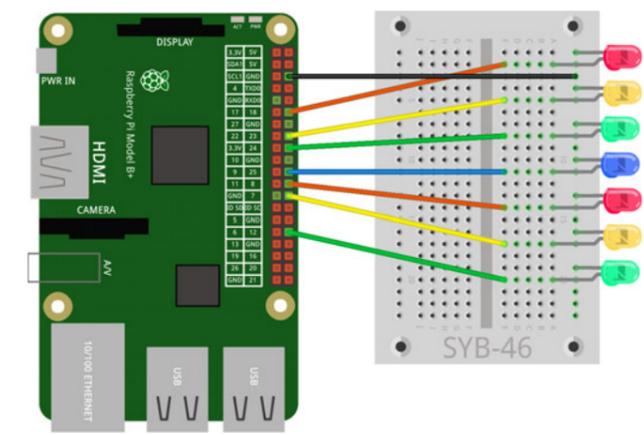
Sieben LEDs blinken in zufälliger Reihenfolge, wobei auch mehrere gleichzeitig eingeschaltet sein können. Das funktioniert so: Immer abwechselnd wird eine zufällig ausgewählte LED eingeschaltet und dann eine wiederum zufällig ausgewählte LED ausgeschaltet.

INFO:

Gemeinhin denkt man, in einem Programm könne nichts zufällig geschehen - wie also kann ein Programm dann in der Lage sein, zufällige Zahlen zu generieren? Teilt man eine große Primzahl durch irgendeinen Wert, ergeben sich ab der x-ten Nachkommastelle Zahlen, die kaum noch vorhersehbar sind. Sie ändern sich auch ohne jede Regelmäßigkeit, wenn man den Divisor regelmäßig erhöht. Dieses Ergebnis ist zwar scheinbar zufällig, lässt sich aber durch ein identisches Programm oder den mehrfachen Aufruf des gleichen Programms jederzeit reproduzieren. Nimmt man jetzt aber eine aus einigen dieser Ziffern zusammengesetzte Zahl und teilt sie wiederum durch eine Zahl, die sich aus der aktuellen Uhrzeitsekunde oder dem Inhalt einer beliebigen Speicherstelle des Computers ergibt, kommt ein Ergebnis heraus, das sich nicht reproduzieren lässt und daher als Zufallszahl bezeichnet wird.

Bauteile

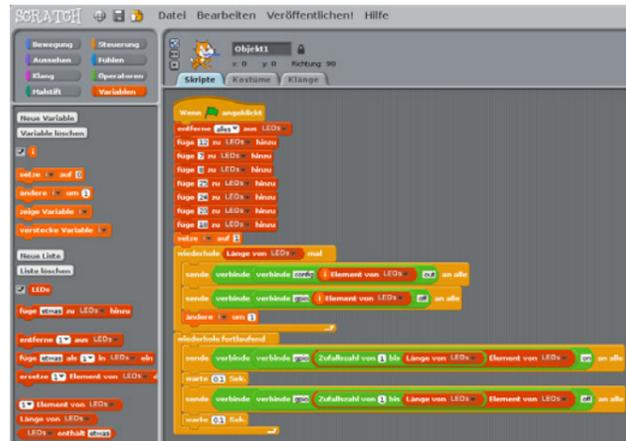
- 1 Steckbrett
- 2 LED rot mit Vorwiderstand
- 2 LED gelb mit Vorwiderstand
- 2 LED grün mit Vorwiderstand
- 1 LED blau mit Vorwiderstand
- 8 GPIO-Verbindungskabel



7 LEDs blinken

Das Programm

Das Programm **04leds** enthält gegenüber den bisherigen Programmen einige neue Elemente.



Das Scratch-Programm 04leds lässt 7 LEDs zufällig blinken

Legen Sie für dieses Programm auf der Blockpalette **Variablen** eine Variable mit Namen **i** und eine Liste mit Namen **LEDs** an. Die Variable **i** wird für einen Schleifenzähler verwendet. Traditionell verwenden Programmierer immer **i, j, k...** für Zähler.

INFO: VARIABLEN UND LISTEN IN SCRATCH



Variablen sind kleine Speicherplätze, in denen sich ein Programm eine Zahl oder irgendetwas anderes merkt. Wenn das Programm beendet wird, werden diese Variablen automatisch wieder geleert. Variablen müssen in Scratch auf der Blockpalette Variablen erst einmal mit einem Klick auf **Neue Variable** angelegt werden, bevor man sie benutzen kann. Anschließend können Sie das Symbol der neu angelegten Variablen aus der Blockpalette in ein dafür vorgesehenes Feld eines Blocks im Programm ziehen. Auf der Blockpalette stehen zusätzlich verschiedene Blöcke zum Auslesen und Verändern der Variablen zur Verfügung.

Eine Liste ist eine besondere Form von Variablen. In solchen Listen können mehrere Werte gespeichert und über die Position innerhalb der Liste adressiert werden. Die Listenelemente werden bei 1 beginnend hochgezählt. Die Nummer des letzten Elementes entspricht der Länge der Liste.

Als erstes wird die Liste **LEDs** komplett geleert, damit sie aus dem letzten Programmlauf keine Daten mehr enthält. Die Inhalte von Variablen und Listen bleiben im Speicher, auch wenn das Programm mit dem roten Stopp-Symbol angehalten und wieder neu gestartet wird.

Anschließend werden nacheinander die sieben Pin-Nummern der für die LEDs verwendeten GPIO-Pins in die Liste eingetragen.

Die Variable **i** wird als Schleifenzähler auf 1 gesetzt und eine Schleife gestartet, die siebenmal durchläuft, entsprechend der Länge der Liste **LEDs**. Scratch bietet einen Block **Länge von...** an, der immer die Länge einer wählbaren Liste enthält.



Um jeden Pin als Ausgang zu definieren und auch auszuschalten, werden die Texte in den **sende ... an alle** Blöcken für jede LED neu zusammengebaut. Der Block **verbinde ...** von der grünen Blockpalette **Operatoren** verbindet zwei beliebige Zeichenketten zu einer. Zwei geschachtelte **verbinde ...** Blöcke verbinden drei Zeichenketten, das Wort **config**, die Nummer des dem aktuellen Schleifenzähler entsprechenden GPIO-Pins und das Wort **out**, um die Pins als Ausgänge zu definieren. Auf die gleiche Weise werden mit **gpio**, der Pinnummer und **off** die Pins ausgeschaltet.

Danach wird der Schleifenzähler um 1 erhöht und der nächste Pin als Ausgang definiert und ausgeschaltet.

Zufallszahl von 1 bis 10

Danach startet eine **wiederhole fortlaufend** Schleife, die eine zufällig ausgewählte LED einschaltet und nach 0,1 Sekunden eine zufällig ausgewählte LED ausschaltet. Auch hier werden die Texte in den **sende ... an alle** Blöcken jedes Mal neu gebaut. Ein Block **Zufallszahl von ... bis...** von der grünen Blockpalette **Operatoren** wählt eine zufällige LED aus. Die Zufallszahl muss dabei zwischen 1 und der Länge der Liste **LEDs**, in unserem Fall **7** liegen.

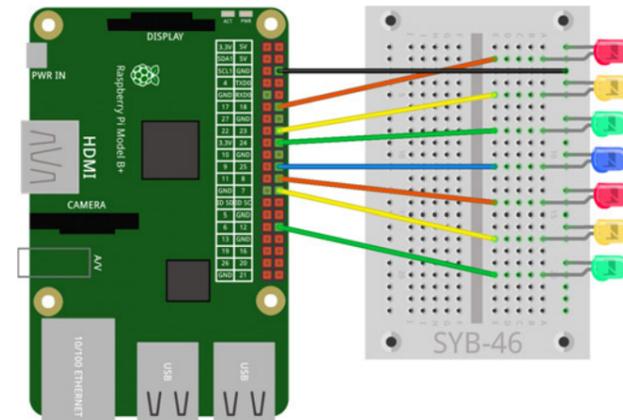
Da die einzuschaltende und die auszuschaltende LED zufällig und unabhängig voneinander ausgewählt werden, können unterschiedlich viele LEDs gleichzeitig eingeschaltet sein. Es ist sogar möglich, dass sich in einem Schleifendurchlauf gar nichts verändert.

LAUFLICHT MIT EINSTELLBARER GESCHWINDIGKEIT

Der Schaltungsaufbau für dieses Experiment entspricht dem vorhergehenden, nur blinken die LEDs nicht zufällig, sondern der Reihe nach abwechselnd. Der Lauflichteffekt entsteht dadurch, dass eine LED eine kurze Zeit leuchtet und wenn sie ausgeschaltet wird, sofort die LED direkt daneben eingeschaltet wird. Nachdem die letzte LED in der Reihe ausgeschaltet wurde, wird die erste wieder eingeschaltet.

Bauteile

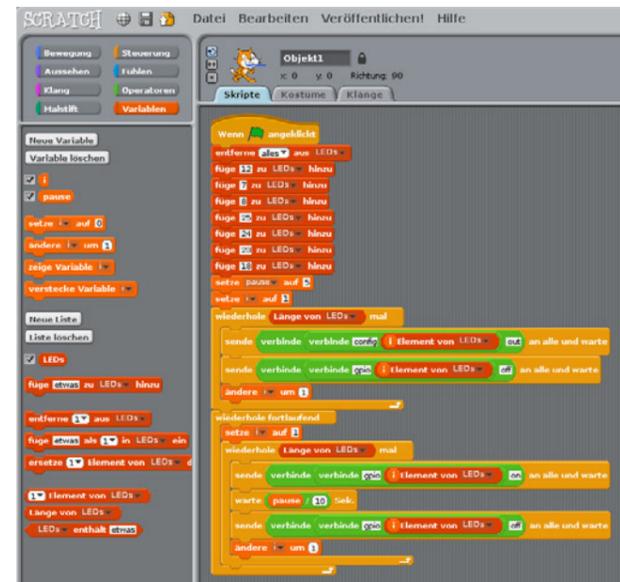
- 1 Steckbrett
- 2 LED rot mit Vorwiderstand
- 2 LED gelb mit Vorwiderstand
- 2 LED grün mit Vorwiderstand
- 1 LED blau mit Vorwiderstand
- 8 GPIO-Verbindungskabel



7 LEDs als Lauflicht

Das Programm

Das Programm **05lauflicht** lässt die LEDs als Lauflicht leuchten. Die Geschwindigkeit lässt sich interaktiv einstellen, während das Programm läuft. Dazu verwendet das Programm eine zusätzliche Variable **pause**, die die Pause zwischen zwei Schaltvorgängen angibt.



Das Scratch-Programm 05lauflicht lässt 7 LEDs als Lauflicht leuchten

Sie können das Programm aus dem vorhergehenden umbauen, da der Anfang weitgehend gleich ist.

Ein Block **setze pause auf 5** von der Blockpalette **Variablen** setzt die Pause auf 5, was im Programm einer halben Sekunde entspricht.

Eine Endlosschleife schaltet nacheinander die LEDs ein und wieder aus. Da in Scratch nur Ganzzahlen als Variablenwerte interaktiv einstellbar sind, dividiert das Programm den Wert der Variablen **pause** durch 10. Verwenden Sie dazu den Block **.../...** von der Blockpalette **Operatoren**. Ziehen Sie das Variablensymbol **pause** von der Blockpalette **Variablen** in das linke Feld des Operators und tragen Sie rechts 10 ein.

Um den Wert der Variable im Programm interaktiv ändern zu können, schalten Sie das Häkchen links neben dem Variablensymbol auf der Blockpalette **Variablen** ein. Damit erscheint die Variable auf der Bühne und zeigt dort immer den aktuellen Wert.

Klicken Sie mit der rechten Maustaste auf die Variable auf der Bühne und wählen Sie im Kontextmenu **Regler**. Damit erscheint ein Regler, mit dem der Variablenwert eingestellt werden kann.



Klicken Sie dann noch auf **Reglerbereich festlegen** und setzen den Maximalwert auf 10, was beim späteren Teilen durch 10 einer Sekunde Pause entspricht.



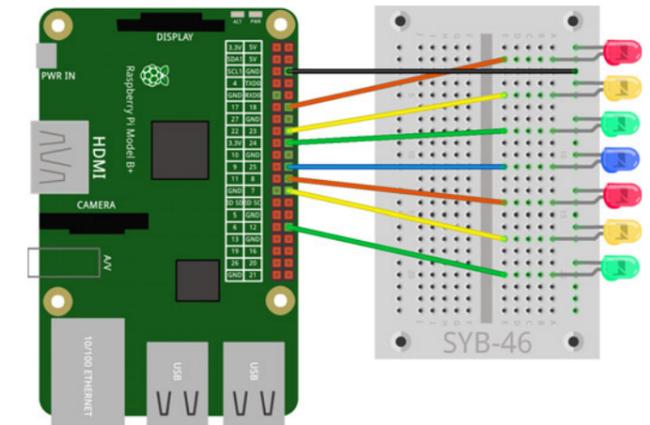
Starten Sie das Programm. Automatisch blinken die LEDs nacheinander auf. Jetzt können Sie mit dem Schieberegler bei der Variable **pause** auf der Bühne die Geschwindigkeit einstellen.

LICHTSTEUERPULT IN SCRATCH

Ein einfaches Lichtsteuerpult schaltet die LEDs durch Klicks in Scratch. Dabei zeigt das Scratch-Steuerpult auch den Zustand der LEDs an.

Bauteile

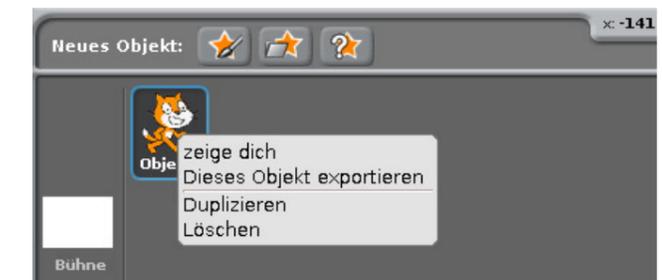
- 1 Steckbrett
- 2 LED rot mit Vorwiderstand
- 2 LED gelb mit Vorwiderstand
- 2 LED grün mit Vorwiderstand
- 1 LED blau mit Vorwiderstand
- 8 GPIO-Verbindungskabel



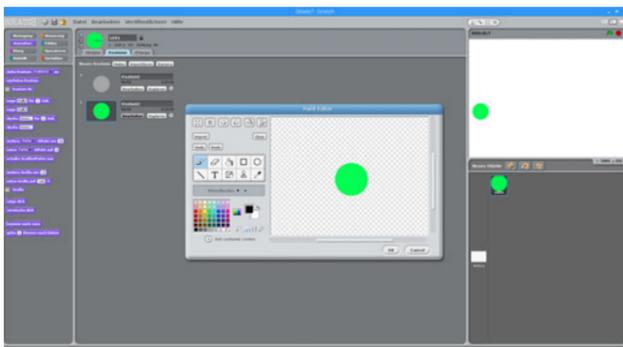
7 LEDs mit einem Steuerpult steuern

Das Programm

Die Scratch-Bühne, auf der bis jetzt nur die Katze zu sehen ist, kann deutlich mehr als nur Variablen anzeigen. Hier können Sie grafische Objekte anlegen, die sich bewegen oder auf Interaktion mit dem Benutzer reagieren. Jedes Programm kann mehrere Objekte enthalten. Im Programm **06leds7** verfügt jedes Objekt über eigene Programmblöcke. Löschen Sie als erstes die Katze, die in diesem Programm nicht gebraucht wird. Klicken Sie dazu mit der rechten Maustaste auf die Katze im Objektfenster unten rechts und wählen Sie im Kontextmenu **Löschen**.



Legen Sie jetzt ein neues Objekt für die erste LED an. Wenn dieses komplett fertig gestaltet und programmiert ist, können Sie es für die anderen LEDs duplizieren und nur noch geringfügig anpassen. Sie können dieses Objekt direkt in Scratch malen. Klicken Sie dazu auf das Symbol **Neues Objekt malen** im Objektfenster. Scratch enthält ein einfaches Malprogramm, mit dem Sie einen grauen Kreis als neues Objekt malen. Geben Sie diesem Objekt im Hauptfenster von Scratch ganz oben den Namen **LED1**.



Jedes Objekt in Scratch kann über so genannte Kostüme sein Aussehen verändern. Wählen Sie das Objekt **LED1** aus und schalten im Skriptfenster auf die Registerkarte **Kostüme**. Kopieren Sie das vorhandene Kostüm mit einem Klick auf **Kopieren**.

Bearbeiten Sie das zweite Kostüm mit dem Malprogramm. Füllen Sie die graue Fläche mit Grün, um die eingeschaltete LED darzustellen. Das Objekt **LED1** bekommt eigene Programmblöcke. Wenn das grüne Fähnchen angeklickt wird, wird zunächst der Pin 12 als Ausgang eingerichtet und ausgeschaltet. Zusätzlich soll auch das Objekt, auf dem Bildschirm ausgeschaltet, grau erscheinen. Hängen Sie dazu an den Block **Wenn grünes Fähnchen angeklickt einen Block Ziehe Kostüm... an** von der Blackpalette **Aussehen**. Wählen Sie im Auswahlfeld dieses Blocks das **Kostüm1**.

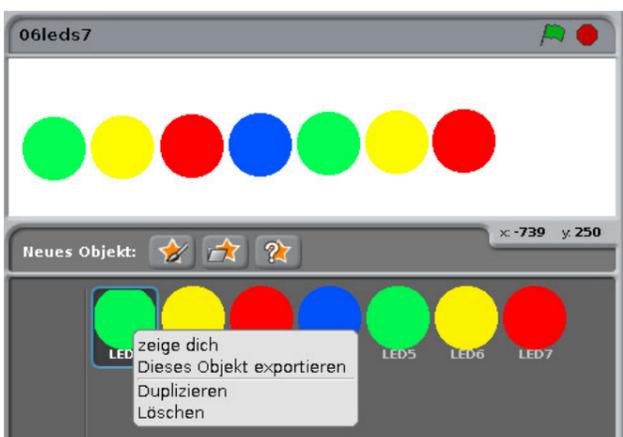


Wird das Objekt selbst angeklickt, soll es auf dem Bildschirm farbig aufleuchten. Gleichzeitig soll auch die entsprechende LED eingeschaltet werden. Ein weiterer Klick soll die LED und auch das Bildschirmobjekt wieder ausschalten.

Verwenden Sie dazu den Block **Wenn LED1 angeklickt** von der Registerkarte **Steuerung**. Beim Anklicken wird zuerst das Kostüm gewechselt und anschließend geprüft, ob aktuell das farbige **Kostüm2** für die eingeschaltete LED zu sehen ist.

In diesem Fall wird die LED am Pin 12 eingeschaltet. Ist dagegen das **Kostüm1** für die ausgeschaltete LED zu sehen, wird die LED am Pin 12 ausgeschaltet.

Nachdem alle Programmblöcke fertig gestellt sind, duplizieren Sie das Objekt **LED1** mit einem Rechtsklick im Objektfenster. Benennen Sie das neue Objekt in **LED2** um.



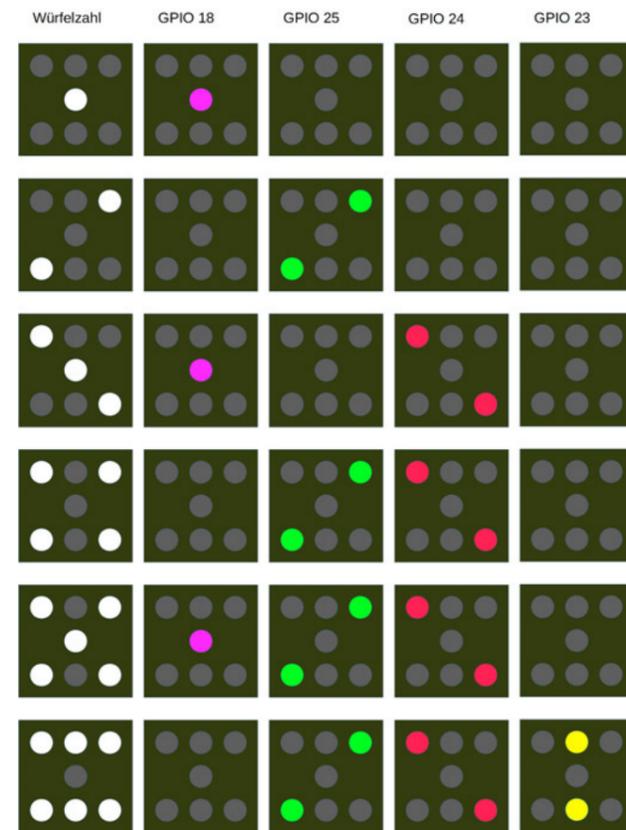
Bearbeiten Sie das **Kostüm2** im Malprogramm. Füllen Sie hier den Kreis mit gelber Farbe. Ändern Sie anschließend im Skriptblock noch den verwendeten GPIO-Pin auf 7. Duplizieren Sie weitere Blöcke, ändern sich die Farben des **Kostüm2** auf die Farben der jeweiligen LEDs und die Pin-Nummern entsprechend.

Starten Sie das Programm mit einem Klick auf das grüne Fähnchen. Alle LEDs sind aus, die Objekte erscheinen in grau. Klicken Sie auf eines der Objekte, erscheint dies in Farbe und die zugehörige LED leuchtet bis zum nächsten Anklicken. So können Sie alle LEDs einzeln ein und ausschalten.

LED-WÜRFEL

Die typischen Spielwürfel, die 1-6 Augen zeigen, kennt jeder und hat jeder zu Hause. Wesentlich cooler ist ein elektronisch gesteuerter Würfel, der mit einem Knetekontakt gesteuert die Augen leuchten lässt - und dann aber nicht einfach 1-6 LEDs in einer Reihe, sondern in der Anordnung eines Spielwürfels.

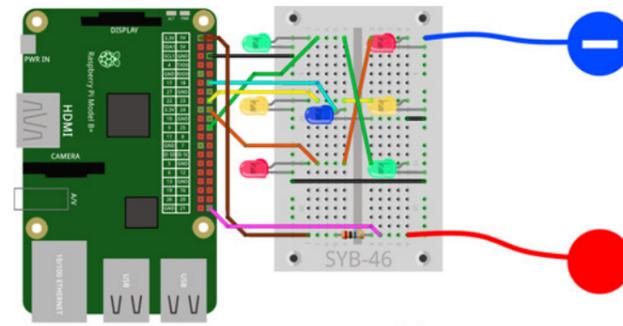
Spielwürfel haben Augen in der typischen quadratischen Anordnung, wozu man 7 LEDs braucht, die in vier Gruppen gesteuert werden, wie die Abbildung zeigt. Für die Ansteuerung der LEDs brauchen wir nur 4 statt 7 GPIO-Pins, da ein Würfel zur Darstellung gerader Zahlen die Augen paarweise nutzt.



Die Würfelaugensymbole mit LED-Gruppen dargestellt

Bauteile

- 1 Steckbrett
- 2 LED rot mit Vorwiderstand
- 2 LED gelb mit Vorwiderstand
- 2 LED grün mit Vorwiderstand
- 1 LED blau mit Vorwiderstand
- 8 GPIO-Verbindungskabel
- 2 Knetkontakte

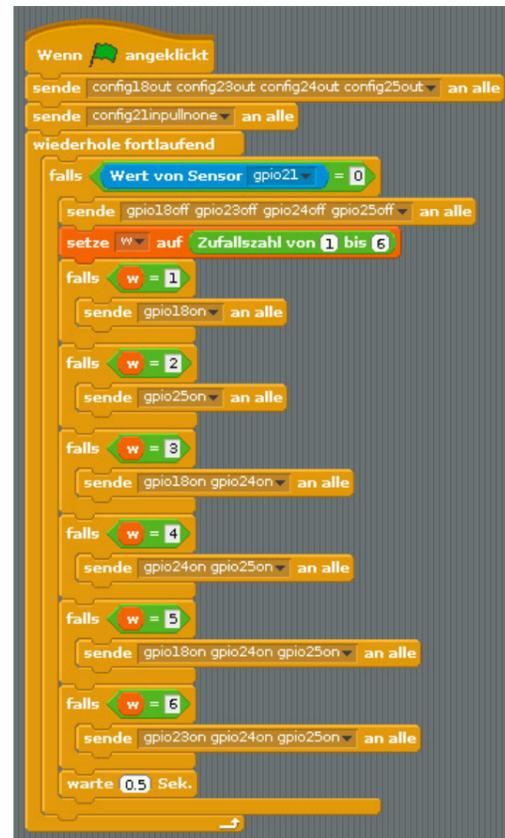


Würfel mit 7 LEDs über Sensorkontakt steuern

Für gleichzeitig geschaltete LEDs verwenden wir gleiche Farben. Für diese Schaltung benötigen Sie einige Drahtbrücken aus Schaltdraht. Schneiden Sie den Draht mit einem kleinen Seitenschneider auf die passenden Längen ab. Um die Drähte besser in die Steckplatine stecken zu können, empfiehlt es sich, die Drähte leicht schräg abzuschneiden, sodass eine Art Keil entsteht. Entfernen Sie dann an beiden Enden mit einem scharfen Messer auf etwa 5 mm Länge die Isolierung.

Das Programm

Das Programm **07wuerfel** simuliert einen Spielwürfel. Alle LEDs sind am Anfang aus. Berühren Sie dann kurz die beiden Knetkontakte, um zu würfeln. Wenn Sie wieder loslassen, bleibt die gewürfelte Zahl angezeigt, bis die Knetkontakte wieder berührt werden.



Das Programm **07wuerfel** simuliert einen Spielwürfel

Wenn der Spieler auf das grüne Fähnchen klickt, werden als erstes die fünf verwendeten GPIO-Pins initialisiert. Wir verwenden hier die Pins 18, 23, 24, 25 als Ausgänge für die LEDs und den Pin 21 als Eingang für den Knetekontakt.

Danach beginnt die Endlosschleife, die darauf wartet, dass die Knetkontakte berührt werden. Dazu wird in jedem Durchlauf der GPIO-Pin 21 abgefragt.

Wenn dieser GPIO-Pin den Wert 0 hat, werden die Anweisungen innerhalb des **falls**-Blocks ausgeführt. Als Erstes schaltet das Programm die vier LEDs aus. Am Anfang sind die LEDs alle aus, aber später bleibt ein ange-

zeigtes Würfelergebnis so lange bestehen, bis wieder die Knetkontakte berührt werden.

Danach wird eine zufällige Zahl zwischen 1 und 6 erzeugt und in der Variable **w** auf der Blackpalette **Variablen** ein, wird diese Variable automatisch auf der Bühne bei der Katze in einem kleinen orangefarbenen Feld angezeigt. So sieht man hier immer die gewürfelte Zahl und kann leicht kontrollieren, ob die LEDs funktionieren. Dieses Zahlenfeld ist sehr klein. Klicken Sie mit der rechten Maustaste darauf und wählen Sie im Menü **Großanzeige**. Dann ist die Zahl besser zu erkennen.

Nachdem die Zahl gewürfelt wurde, folgen sechs **falls**-Blöcke für jeden möglichen Würfelwert. Jeder dieser Blöcke schaltet wenn eine bestimmte Zahl gewürfelt wurde, die entsprechende Kombination von LEDs ein.

Unabhängig vom Würfelergebnis soll das Programm immer nachdem dem Knetekontakt neu gewürfelt wurde, noch eine halbe Sekunde warten, um zu vermeiden, dass durch sogenanntes Tastenprellen kurz hintereinander zwei verschiedene Würfelfaktionen ausgelöst werden. Dies erledigt ein Block **warte 0.5 Sek.** nach dem letzten **falls**-Block. Dieser wird in jedem Fall abgearbeitet.

Alle vorgestellten Schaltungen und Programme wurden mit der größtmöglichen Sorgfalt entwickelt, geprüft und getestet. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag und Autor haften in Fällen des Vorsatzes oder der groben Fahrlässigkeit nach den gesetzlichen Bestimmungen. Im Übrigen haften Verlag und Autor nur nach dem Produkthaftungsgesetz wegen der Verletzung des Lebens, des Körpers oder der Gesundheit oder wegen der schuldhafte Verletzung wesentlicher Vertragspflichten. Der Schadensersatzanspruch für die Verletzung wesentlicher Vertragspflichten ist auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht ein Fall der zwingenden Haftung nach dem Produkthaftungsgesetz gegeben ist.

Achtung! Augenschutz und LEDs:

Blicken Sie nicht aus geringer Entfernung direkt in eine LED, denn ein direkter Blick kann Netzhautschäden verursachen! Dies gilt besonders für helle LEDs im klaren Gehäuse sowie in besonderem Maße für Power-LEDs. Bei weißen, blauen, violetten und ultravioletten LEDs gibt die scheinbare Helligkeit einen falschen Eindruck von der tatsächlichen Gefahr für Ihre Augen. Besondere Vorsicht ist bei der Verwendung von Sammellinsen geboten. Betreiben Sie die LEDs so wie in der Anleitung vorgesehen, nicht aber mit größeren Strömen.

Liebe Kunden!

Dieses Produkt wurde in Übereinstimmung mit den geltenden europäischen Richtlinien hergestellt und trägt daher das CE-Zeichen. Der bestimmungsgemäße Gebrauch ist in der beiliegenden Anleitung beschrieben.

Bei jeder anderen Nutzung oder Veränderung des Produktes sind allein Sie für die Einhaltung der geltenden Regeln verantwortlich. Bauen Sie die Schaltungen deshalb genau so auf, wie es in der Anleitung beschrieben wird. Das Produkt darf nur zusammen mit dieser Anleitung weitergegeben werden.

Das Symbol der durchkreuzten Mülltonne bedeutet, dass dieses Produkt getrennt vom Hausmüll als Elektroschrott dem Recycling zugeführt werden muss. Wo Sie die nächstgelegene kostenlose Annahmestelle finden, sagt Ihnen Ihre kommunale Verwaltung.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Das Erstellen und Verbreiten von Kopien auf Papier, auf Datenträgern oder im Internet, insbesondere als PDF, ist nur mit ausdrücklicher Genehmigung des Verlags gestattet und wird widrigenfalls strafrechtlich verfolgt.

Die meisten Produktbezeichnungen von Hard- und Software sowie Firmennamen und Firmenlogos, die in diesem Werk genannt werden, sind in der Regel gleichzeitig auch eingetragene Warenzeichen und sollten als solche betrachtet werden. Der Verlag folgt bei den Produktbezeichnungen im Wesentlichen den Schreibweisen der Hersteller.

Autor: Christian Immler GTIN: 4019631150431 N° 1891587



Für Kinder unter 14 Jahren nicht geeignet!

MAKERFACTORY distributed by Conrad Electronic SE Klaus-Conrad-Str. 11 92240 Hirschau www.makerfactory.com

© 2019 Franzis Verlag GmbH Richard-Reitzner-Allee 2 D-85540 Haar, Germany 2019/01